Forecasting Monthly Room Occupancy of Chelsea Hotel Using ARIMA

Lilian

11-02-2023

```r
#import libraries
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

#Data Wrangling & Exploratory Data Analysis

```r
# IMPORT DATA
rm <- read.csv("C:/Users/LILIAN/Desktop/Chelsea Room Occupancy.csv")
```

```r
#To voew few columns
head(rm)
```

```
##   N.A    Date Total_Rooms  X X.1 X.2 X.3 X.4 X.5
## 1   1 01-2013        2048 NA  NA  NA  NA  NA  NA
## 2   2 02-2013        2755 NA  NA  NA  NA  NA  NA
## 3   3 03-2013        2113 NA  NA  NA  NA  NA  NA
## 4   4 04-2013        2036 NA  NA  NA  NA  NA  NA
## 5   5 05-2013        2331 NA  NA  NA  NA  NA  NA
## 6   6 06-2013        2769 NA  NA  NA  NA  NA  NA
```

```r
#check the class of the data set
class(rm)
```

```
## [1] "data.frame"
```

THE DATASET NEEDS TO BE CONVERTED TO TIMESERIES DATA

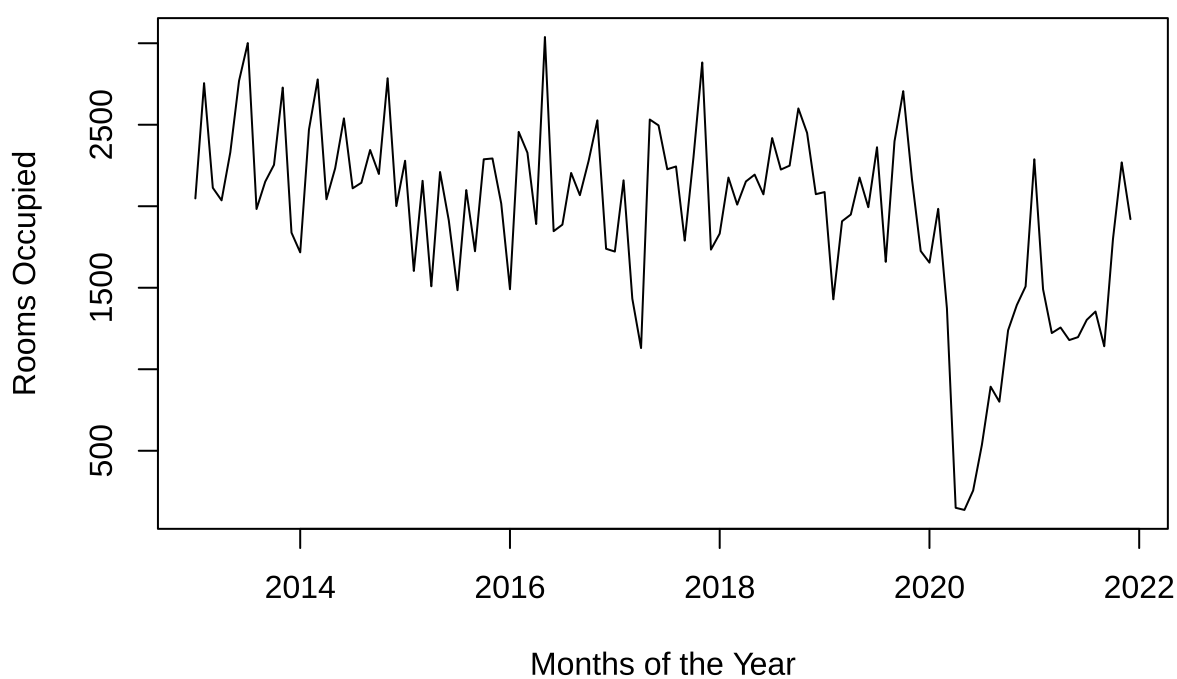
```r
#convert the data into a time series data
rm_ts <- ts(rm$Total_Rooms, frequency = 12, start = c(2013,1))
rm_ts
```

```
##       Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
## 2013 2048 2755 2113 2036 2331 2769 3001 1983 2151 2254 2728 1838
## 2014 1717 2470 2778 2043 2233 2539 2110 2144 2345 2198 2785 2001
## 2015 2279 1603 2156 1509 2210 1913 1485 2099 1724 2288 2293 2017
## 2016 1491 2456 2328 1891 3038 1847 1887 2204 2068 2278 2527 1739
## 2017 1722 2159 1433 1130 2532 2496 2227 2244 1789 2301 2882 1734
## 2018 1832 2176 2010 2152 2194 2073 2418 2225 2249 2600 2450 2074
## 2019 2087 1429 1908 1949 2176 1994 2362 1659 2398 2706 2166 1725
## 2020 1654 1984 1373  150  137  256  536  893  801 1239 1394 1508
## 2021 2288 1491 1222 1256 1179 1197 1303 1354 1141 1799 2269 1921
```

```r
#confirm the class
class(rm_ts)
```

```
## [1] "ts"
```

```r
#TO CHECK FOR STATIONARITY OF THE DATA BY PLOT
plot(rm_ts, xlab='Months of the Year', ylab = 'Rooms Occupied')
```



THE CHART SHOWS THE DATA IS NOT STATIONARY. SO WE TAKE THE LOG OF THE DATA TO MEKE IT STATIONARY

```r
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library(forecast) #for time series prediction
```

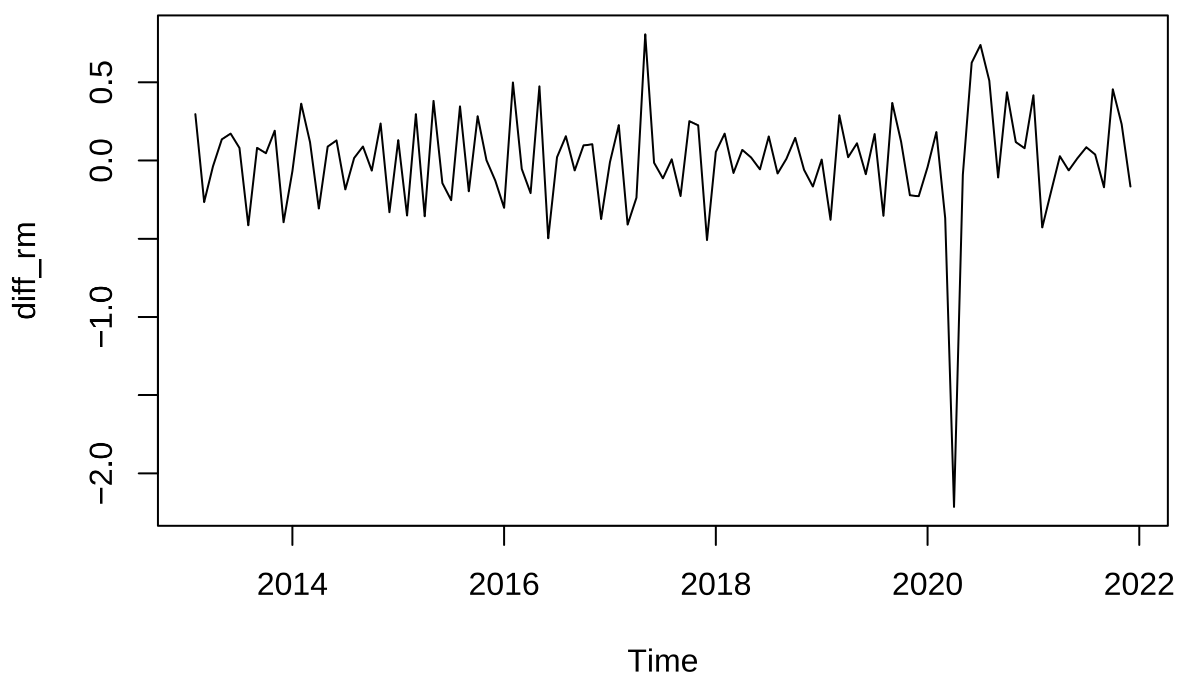
```r
# WE USE THE A. DICKEY FULLER TEST TO CONFIRM STATIONARITY #the data is #stationery
adf.test(rm_ts)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  rm_ts
## Dickey-Fuller = -3.224, Lag order = 4, p-value = 0.08751
## alternative hypothesis: stationary
```

THE P-VALUE IS > THAN 0.05 WHICH IS NOT ACCEPTABLE. DATA IS NOT STATIONARY

```r
#take the difference
#rm_diff = plot(diff(rm_ts),ylab='Rooms Occupied')

#TAKE THE LOG AND check if the log value is stationery
diff_rm = diff(log(rm_ts))
plot(diff_rm)
```



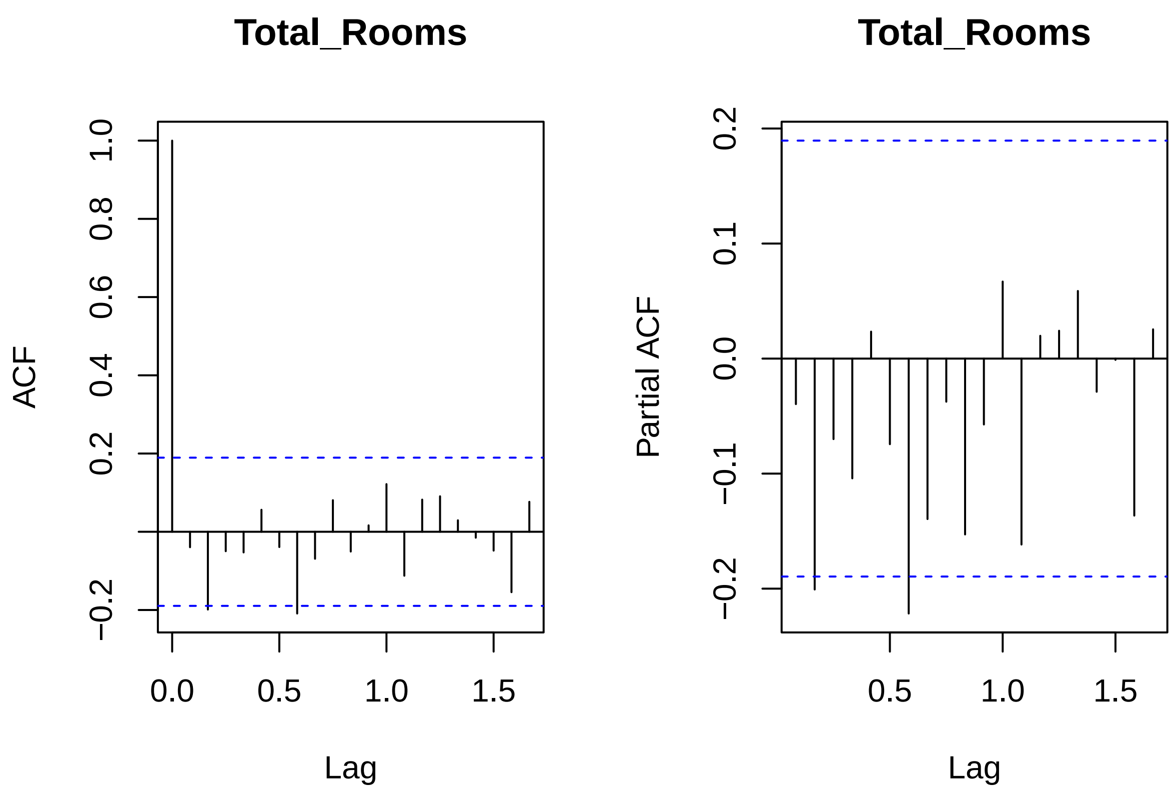
```r
#WE USE THE A. DICKEY FULLER TEST TO CONFIRM STATIONARITY
adf.test(diff_rm) #the data is stationery
```

```
## Warning in adf.test(diff_rm): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff_rm
## Dickey-Fuller = -5.1439, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

THE P-VALUE IS SIGNIFICANT AND IT IS RIGH FOR THE TIMESERIES ANALYSIS

```r
#check the ACF and PACF
par(mfrow=c(1,2))
acf(as.ts(diff_rm), main="Total_Rooms")
pacf(as.ts(diff_rm), main="Total_Rooms")
```



THE CHART SHOWS THAT ONLY FEW LAGS ARE OUTSIDE THE BLUE LINES WHICH SHOWS THAT WE CAN PROCEED WITH PREDICTION

```r
#TO RUN THE AUTO ARIMA WHICH WILL GIVE US THE BEST MODEL FOR ARIMA ANALYSIS
rm_model <- auto.arima(diff(diff_rm), trace=TRUE)
```

```
##
##  ARIMA(2,0,2)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)            with non-zero mean : 154.0038
##  ARIMA(1,0,0)(1,0,0)[12] with non-zero mean : 136.3943
##  ARIMA(0,0,1)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)            with zero mean     : 151.9341
##  ARIMA(1,0,0)            with non-zero mean : 135.7423
##  ARIMA(1,0,0)(0,0,1)[12] with non-zero mean : 136.5387
##  ARIMA(1,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,0)            with non-zero mean : 119.6602
##  ARIMA(2,0,0)(1,0,0)[12] with non-zero mean : 121.0393
##  ARIMA(2,0,0)(0,0,1)[12] with non-zero mean : 121.0937
##  ARIMA(2,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(3,0,0)            with non-zero mean : 114.297
##  ARIMA(3,0,0)(1,0,0)[12] with non-zero mean : 115.4222
##  ARIMA(3,0,0)(0,0,1)[12] with non-zero mean : 115.4381
##  ARIMA(3,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(4,0,0)            with non-zero mean : 106.2016
##  ARIMA(4,0,0)(1,0,0)[12] with non-zero mean : 105.405
##  ARIMA(4,0,0)(2,0,0)[12] with non-zero mean : 107.4191
##  ARIMA(4,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(4,0,0)(0,0,1)[12] with non-zero mean : 105.63
##  ARIMA(4,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(5,0,0)(1,0,0)[12] with non-zero mean : 105.1597
##  ARIMA(5,0,0)            with non-zero mean : 105.9012
##  ARIMA(5,0,0)(2,0,0)[12] with non-zero mean : 107.2564
##  ARIMA(5,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(5,0,0)(0,0,1)[12] with non-zero mean : 105.3516
##  ARIMA(5,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(5,0,1)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(4,0,1)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(5,0,0)(1,0,0)[12] with zero mean     : 102.818
##  ARIMA(5,0,0)            with zero mean     : 103.6076
##  ARIMA(5,0,0)(2,0,0)[12] with zero mean     : 104.8659
##  ARIMA(5,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(5,0,0)(0,0,1)[12] with zero mean     : 103.01
##  ARIMA(5,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(4,0,0)(1,0,0)[12] with zero mean     : 103.1106
##  ARIMA(5,0,1)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(4,0,1)(1,0,0)[12] with zero mean     : Inf
##
##  Best model: ARIMA(5,0,0)(1,0,0)[12] with zero mean
```

THE ABOVE TABLE SHOWS THE ARIMA 5,0,0 ANS 1,0,0 ARE BEST FIT FOR OUR MODEL BECAUSE IT HAS THE LOWEST 102.818 AICAIKE INFORMATION CRITERIA (AIC)

```r
rm_model = arima(diff_rm,order = c(1,0,0))
rm_model
```

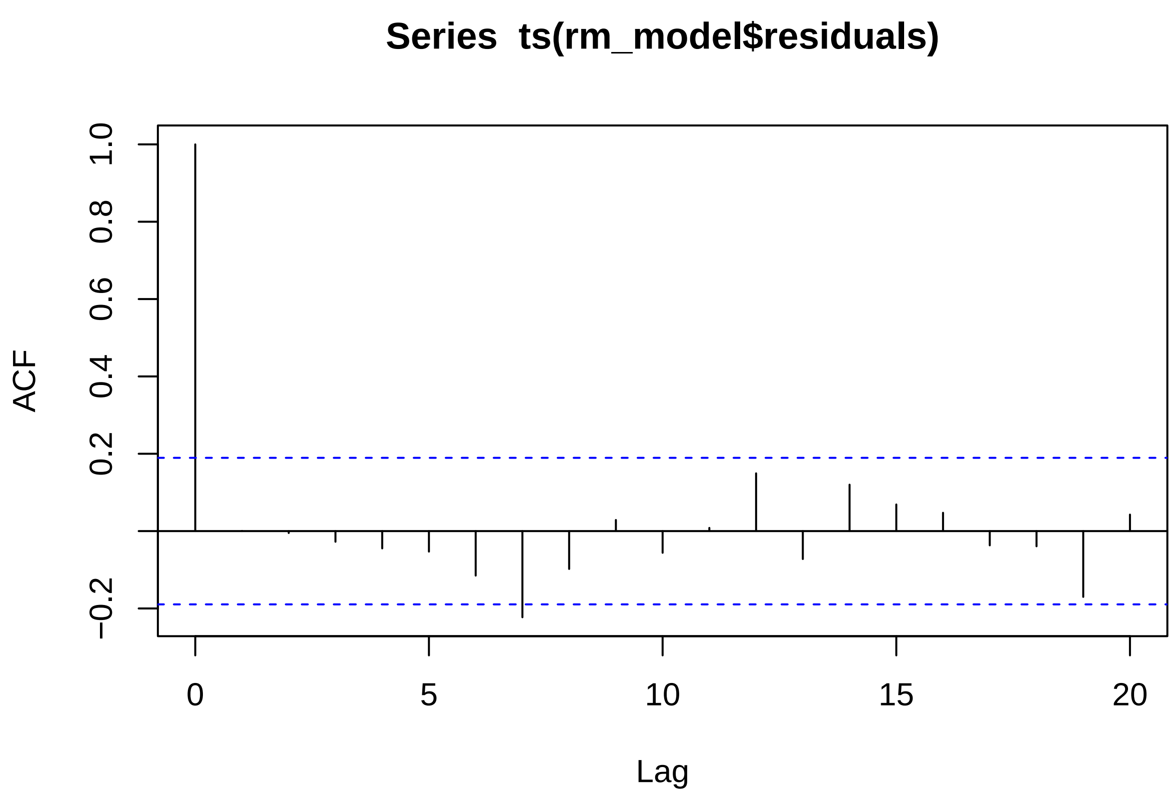
```
##
## Call:
## arima(x = diff_rm, order = c(1, 0, 0))
##
```

```
## Coefficients:
##           ar1  intercept
##       -0.0396    -0.0006
## s.e.   0.0966     0.0316
##
## sigma^2 estimated as 0.1151:  log likelihood = -36.15,  aic = 78.3
```

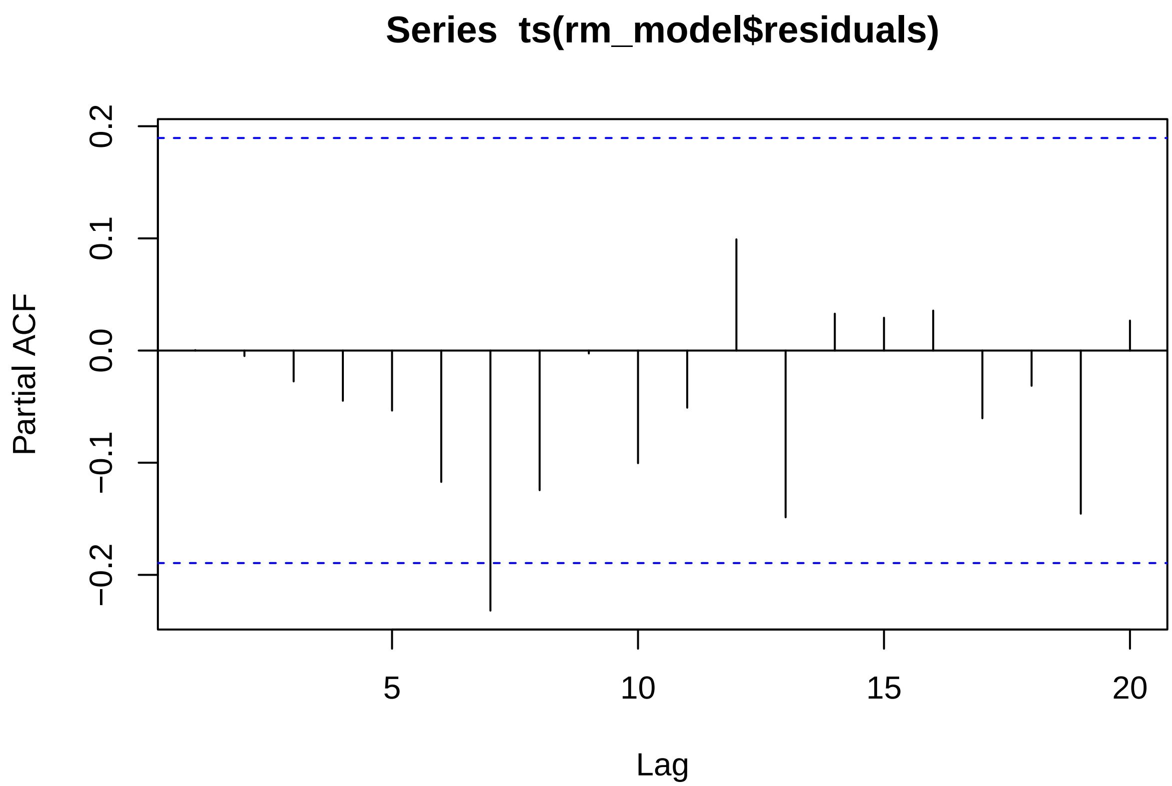
```
rm_model = arima(diff_rm,order = c(5,0,0))
rm_model
```

```
##
## Call:
## arima(x = diff_rm, order = c(5, 0, 0))
##
## Coefficients:
##           ar1      ar2      ar3      ar4     ar5  intercept
##       -0.0647  -0.2220  -0.0706  -0.1004  0.0235    -0.0013
## s.e.   0.0968   0.0963   0.0988   0.0963  0.0964     0.0224
##
## sigma^2 estimated as 0.1086:  log likelihood = -33.11,  aic = 80.21
```

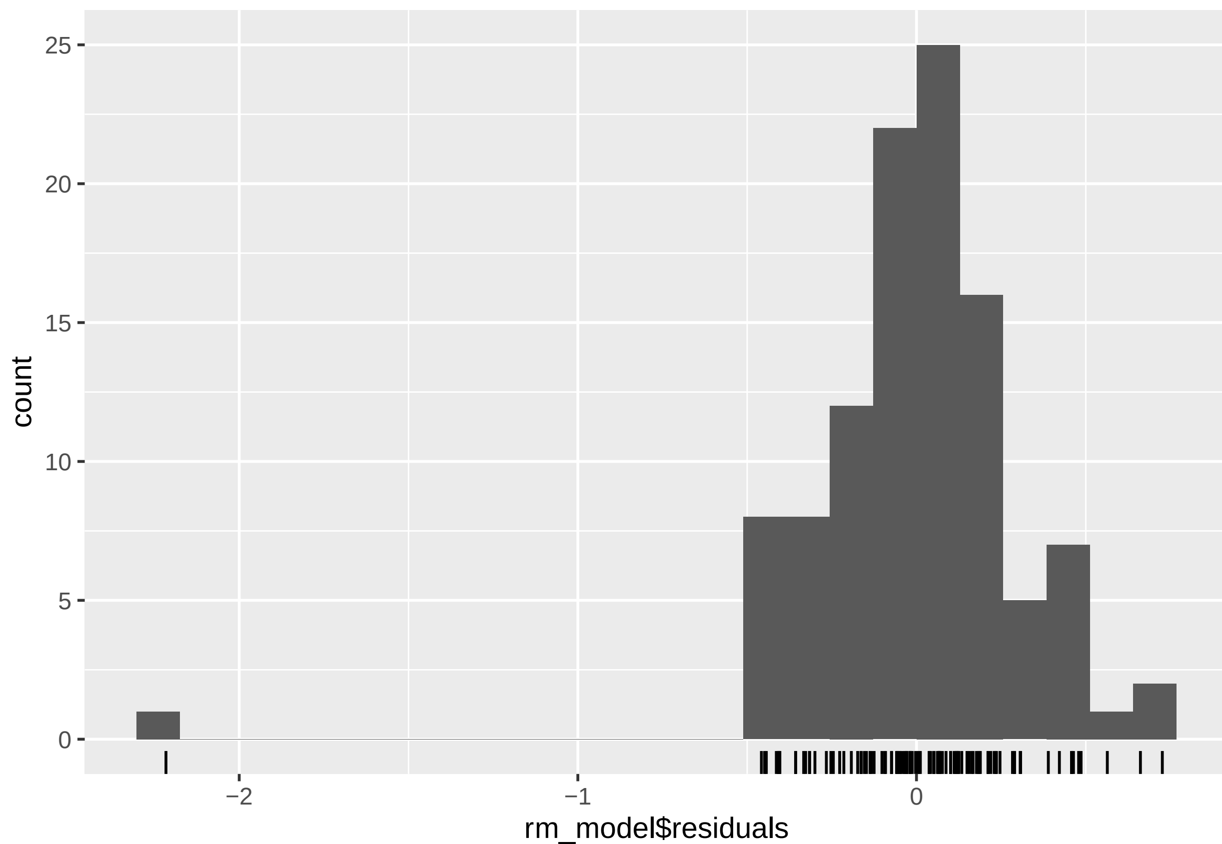
```
#to check for the residual
acf(ts(rm_model$residuals))
```



```
pacf(ts(rm_model$residuals))
```




```
# plot the graph of histogram to check if my residual are normally distributed
gghistogram(rm_model$residuals)
```



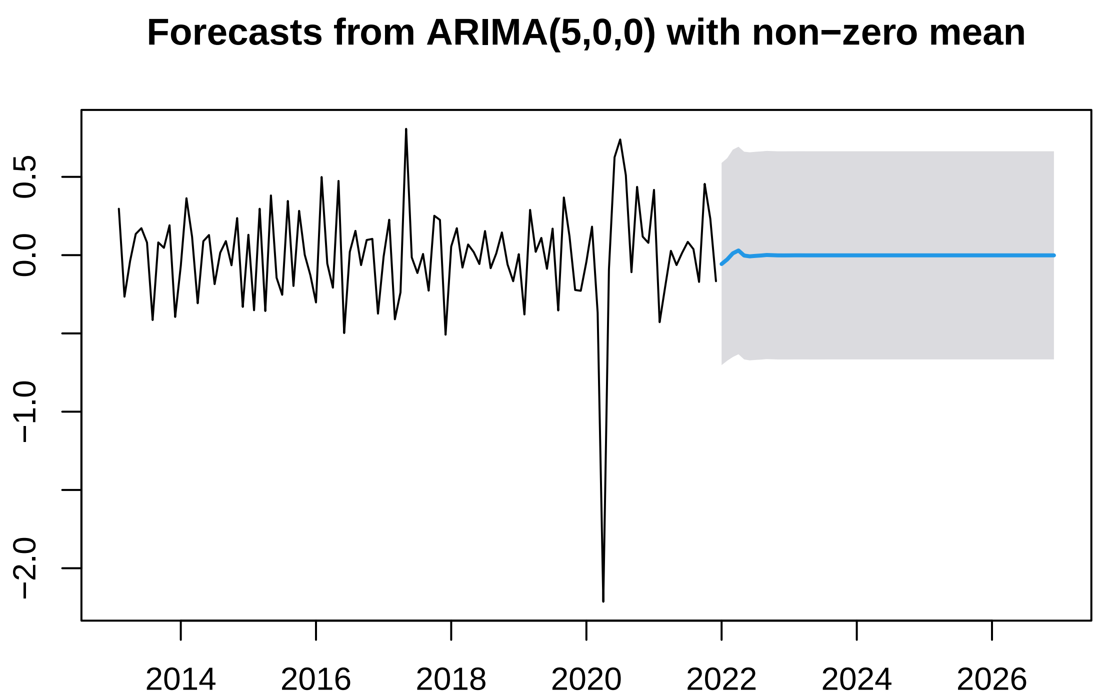
IT IS NORMALLY DISTRIBUTED

```r
#WE FORECAST THE TS DATA FOR 5 YEARS AT 95% CONFIDENCE LEVEL
rm_forecast = forecast(rm_model, level = c(95), h = 5*12)#confidence level 95%
rm_forecast
```

```
##          Point Forecast      Lo 95     Hi 95
## Jan 2022  -0.0566156289 -0.7024174 0.5891862
## Feb 2022  -0.0272909160 -0.6744433 0.6198615
## Mar 2022   0.0116801734 -0.6505816 0.6739420
## Apr 2022   0.0296637649 -0.6331571 0.6924846
## May 2022  -0.0026110141 -0.6660613 0.6608393
## Jun 2022  -0.0076300357 -0.6721225 0.6568624
## Jul 2022  -0.0046338777 -0.6693998 0.6601321
## Aug 2022  -0.0023251391 -0.6671395 0.6624892
## Sep 2022   0.0008769050 -0.6639500 0.6657038
## Oct 2022  -0.0003083313 -0.6651456 0.6645289
## Nov 2022  -0.0015240825 -0.6663617 0.6633135
## Dec 2022  -0.0015697903 -0.6664114 0.6632718
## Jan 2023  -0.0014804976 -0.6663222 0.6633612
## Feb 2023  -0.0011961272 -0.6660378 0.6636456
## Mar 2023  -0.0011369086 -0.6659787 0.6637049
## Apr 2023  -0.0012341301 -0.6660760 0.6636077
## May 2023  -0.0012710993 -0.6661129 0.6635707
## Jun 2023  -0.0012777561 -0.6661196 0.6635641
## Jul 2023  -0.0012615220 -0.6661034 0.6635803
```

```
## Aug 2023  -0.0012473345 -0.6660892 0.6635945
## Sep 2023  -0.0012499580 -0.6660918 0.6635919
## Oct 2023  -0.0012542837 -0.6660961 0.6635876
## Nov 2023  -0.0012562090 -0.6660981 0.6635856
## Dec 2023  -0.0012559820 -0.6660978 0.6635859
## Jan 2024  -0.0012546674 -0.6660965 0.6635872
## Feb 2024  -0.0012542943 -0.6660961 0.6635876
## Mar 2024  -0.0012545346 -0.6660964 0.6635873
## Apr 2024  -0.0012547627 -0.6660966 0.6635871
## May 2024  -0.0012548475 -0.6660967 0.6635870
## Jun 2024  -0.0012547810 -0.6660966 0.6635871
## Jul 2024  -0.0012547175 -0.6660966 0.6635871
## Aug 2024  -0.0012547131 -0.6660966 0.6635871
## Sep 2024  -0.0012547290 -0.6660966 0.6635871
## Oct 2024  -0.0012547421 -0.6660966 0.6635871
## Nov 2024  -0.0012547429 -0.6660966 0.6635871
## Dec 2024  -0.0012547378 -0.6660966 0.6635871
## Jan 2025  -0.0012547353 -0.6660966 0.6635871
## Feb 2025  -0.0012547356 -0.6660966 0.6635871
## Mar 2025  -0.0012547367 -0.6660966 0.6635871
## Apr 2025  -0.0012547373 -0.6660966 0.6635871
## May 2025  -0.0012547371 -0.6660966 0.6635871
## Jun 2025  -0.0012547368 -0.6660966 0.6635871
## Jul 2025  -0.0012547367 -0.6660966 0.6635871
## Aug 2025  -0.0012547368 -0.6660966 0.6635871
## Sep 2025  -0.0012547369 -0.6660966 0.6635871
## Oct 2025  -0.0012547369 -0.6660966 0.6635871
## Nov 2025  -0.0012547368 -0.6660966 0.6635871
## Dec 2025  -0.0012547368 -0.6660966 0.6635871
## Jan 2026  -0.0012547368 -0.6660966 0.6635871
## Feb 2026  -0.0012547368 -0.6660966 0.6635871
## Mar 2026  -0.0012547368 -0.6660966 0.6635871
## Apr 2026  -0.0012547368 -0.6660966 0.6635871
## May 2026  -0.0012547368 -0.6660966 0.6635871
## Jun 2026  -0.0012547368 -0.6660966 0.6635871
## Jul 2026  -0.0012547368 -0.6660966 0.6635871
## Aug 2026  -0.0012547368 -0.6660966 0.6635871
## Sep 2026  -0.0012547368 -0.6660966 0.6635871
## Oct 2026  -0.0012547368 -0.6660966 0.6635871
## Nov 2026  -0.0012547368 -0.6660966 0.6635871
## Dec 2026  -0.0012547368 -0.6660966 0.6635871
```

```
#TO PLOT THE PREDICTED YEAR
plot(rm_forecast)
```



THE ABOVE CHART SHOWS THE LOWER AND HIGHER LIMIT CONFIDENCE LEVEL AROUND THE GREY AREA

```
# 5 predicted values
rm_forecast$mean
```

```
##                Jan           Feb           Mar           Apr           May
## 2022 -0.0566156289 -0.0272909160  0.0116801734  0.0296637649 -0.0026110141
## 2023 -0.0014804976 -0.0011961272 -0.0011369086 -0.0012341301 -0.0012710993
## 2024 -0.0012546674 -0.0012542943 -0.0012545346 -0.0012547627 -0.0012548475
## 2025 -0.0012547353 -0.0012547356 -0.0012547367 -0.0012547373 -0.0012547371
## 2026 -0.0012547368 -0.0012547368 -0.0012547368 -0.0012547368 -0.0012547368
##                Jun           Jul           Aug           Sep           Oct
## 2022 -0.0076300357 -0.0046338777 -0.0023251391  0.0008769050 -0.0003083313
## 2023 -0.0012777561 -0.0012615220 -0.0012473345 -0.0012499580 -0.0012542837
## 2024 -0.0012547810 -0.0012547175 -0.0012547131 -0.0012547290 -0.0012547421
## 2025 -0.0012547368 -0.0012547367 -0.0012547368 -0.0012547369 -0.0012547369
## 2026 -0.0012547368 -0.0012547368 -0.0012547368 -0.0012547368 -0.0012547368
##                Nov           Dec
## 2022 -0.0015240825 -0.0015697903
## 2023 -0.0012562090 -0.0012559820
## 2024 -0.0012547429 -0.0012547378
## 2025 -0.0012547368 -0.0012547368
## 2026 -0.0012547368 -0.0012547368
```

```r
# to validate this forecasting using the box test

Box.test(rm_forecast$resid, lag = 5, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  rm_forecast$resid
## X-squared = 0.63624, df = 5, p-value = 0.9863
```

```r
Box.test(rm_forecast$resid, lag = 9, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  rm_forecast$resid
## X-squared = 9.1822, df = 9, p-value = 0.4206
```

```r
Box.test(rm_forecast$resid, lag = 15, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  rm_forecast$resid
## X-squared = 15.365, df = 15, p-value = 0.4255
```

p-value for box test should be more than 0.05 which means your data is not having any autocorrelation. p-value should not have significant level

```r
# to check for accuracy
accuracy(rm_forecast)
```

```
##                       ME      RMSE       MAE      MPE     MAPE      MASE
## Training set 0.000238127 0.3294968 0.2148934 82.25436 117.6539 0.7085973
##                      ACF1
## Training set 0.0004163456
```